**Program:**

**Main.java**

//Final year project management system

package Projpack;

import java.util.Scanner;

public class Main {

public static void main(String[] args) {

FYproject.*createDatabase*();

FYproject.*createTables*();

Scanner scanner = new Scanner(System.***in***);

System.***out***.println("Welcome to our final year project management system");

while (true) {

System.***out***.println("Select an option:");

System.***out***.println("1. User");

System.***out***.println("2. Professor");

System.***out***.println("3. Exit");

int option = scanner.nextInt();

scanner.nextLine(); // Consume the newline character

switch (option) {

case 1:

*userMenu*();

break;

case 2:

*FacultyMenu*();

break;

case 3:

System.***out***.println("Exiting. Thank You");

System.*exit*(0);

break;

default:

System.***out***.println("Invalid option. Please try again.");

}

}

}

private static void userMenu() {

Scanner scanner = new Scanner(System.***in***);

while (true) {

System.***out***.println("Select any option:");

System.***out***.println("1. View all projects");

System.***out***.println("2. Search by project name");

System.***out***.println("3. Search by student name");

System.***out***.println("4. Exit");

int userOption = scanner.nextInt();

scanner.nextLine(); // Consume the newline character

switch (userOption) {

case 1:

FYproject.*viewallprojects*();

break;

case 2:

System.***out***.print("Enter project name: ");

String projectName = scanner.nextLine();

FYproject.*searchbyprojectname*(projectName);

break;

case 3:

System.***out***.print("Enter student name: ");

String studentName = scanner.nextLine();

FYproject.*searchbystudentname*(studentName);

break;

case 4:

System.***out***.println("Exiting user menu.");

return;

default:

System.***out***.println("Invalid option. Please try again.");

}

}

}

private static void FacultyMenu() {

Scanner scanner = new Scanner(System.***in***);

System.***out***.print("Enter your faculty ID: ");

String facultyId = scanner.nextLine();

if (facultyId.equals("admin")) {

System.***out***.println("Authentication successful!");

while (true) {

System.***out***.println("Select an option:");

System.***out***.println("1. Add Details");

System.***out***.println("2. Delete");

System.***out***.println("3. Search by project name");

System.***out***.println("4. Search by student name");

System.***out***.println("5. Update research paper status of the project");

System.***out***.println("6. Exit");

int facultyOption = scanner.nextInt();

scanner.nextLine(); // Consume the newline character

switch (facultyOption) {

case 1:

System.***out***.print("Enter team ID");

String teamID = scanner.nextLine();;

System.***out***.print("Enter project name");

String projectname = scanner.nextLine();

System.***out***.print("Enter team leader name");

String leadername = scanner.nextLine();

System.***out***.print("Enter team member1 name:");

String teammember1 = scanner.nextLine();

System.***out***.print("Enter team member2 name:");

String teammember2 = scanner.nextLine();

System.***out***.print("Enter team member3 name:");

String teammember3 = scanner.nextLine();

System.***out***.print("Enter year");

String year= scanner.nextLine();

System.***out***.print("Enter research paper status(Published/Not Published)");

String rpstatus = scanner.nextLine();

Faculty.*addDetails*(teamID,projectname,leadername,teammember1, teammember2, teammember3,year,rpstatus);

break;

case 2:

System.***out***.print("Enter team ID to delete");

String team\_ID = scanner.nextLine();

Faculty.*deleteProject*(team\_ID);

break;

case 3:

System.***out***.print("Enter project name: ");

String projectName = scanner.nextLine();

FYproject.*searchbyprojectname*(projectName);

break;

case 4:

System.***out***.print("Enter student name: ");

String studentName = scanner.nextLine();

FYproject.*searchbystudentname*(studentName);

break;

case 5:

System.***out***.print("Enter team ID to update");

String teamid = scanner.nextLine();

System.***out***.print("Enter research paper status (PUBLISHED/NOT PUBLISHED): ");

String researchpaperstatus = scanner.nextLine();

Faculty.*update*(researchpaperstatus,teamid);

break;

case 6:

System.***out***.println("Exiting faculty menu.");

System.*exit*(0);

default:

System.***out***.println("Invalid option. Please try again.");

}

}

} else {

System.***out***.println("Authentication failed. Only college faculty can access this feature.");

}

}

}

**FYproject.java**

package Projpack;

import java.sql.Connection;

import java.sql.DriverManager;

import java.sql.PreparedStatement;

import java.sql.ResultSet;

import java.sql.Statement;

public class FYproject {

public static void createDatabase() {

try {

String url = "jdbc:mysql://localhost:3306/";

String username = "root";

String password = "Tanuvinnu@1234";

Connection conn = DriverManager.getConnection(url, username, password);

Statement stm = conn.createStatement();

System.out.println("Connected Successfully");

// Check if the 'finalyeatprojectDB' database exists

ResultSet resultSet = stm.executeQuery("SHOW DATABASES LIKE 'finalyearprojectDB' ");

if (!resultSet.next()) {

// If ' finalyeatprojectDB ' database doesn't exist, create it

String createDbQuery = "CREATE DATABASE IF NOT EXISTS finalyearprojectDB ";

stm.executeUpdate(createDbQuery);

System.out.println("'finalyearprojectDB' database created successfully.");

} else {

System.out.println("'finalyearprojectDB' database already exists.");

}

// Use the ' finalyeatprojectDB ' database

conn.setCatalog("finalyearprojectDB");

conn.close();

} catch (Exception e) {

e.printStackTrace();

}

}

public static void createTables() {

try {

Connection conn = DBconnect ();

Statement stm = conn.createStatement();

System.out.println("Connected Successfully");

// Check if the ' ProjectDetails ' table exists

ResultSet resultSet = stm.executeQuery("SHOW TABLES LIKE 'ProjectDetails'");

if (!resultSet.next()) {

// If ' ProjectDetails ' table doesn't exist, create it

String createProjectDetailsTableQuery = "CREATE TABLE ProjectDetails (teamId varchar(16) primary key, projectname varchar(55),leadername varchar(15) , teammember1 varchar(25), teammember2 varchar(25) ,teammember3 varchar(25) ,year varchar(20), researchpaperstatus varchar(50) ) ";

stm.executeUpdate(createProjectDetailsTableQuery);

System.out.println("' ProjectDetails ' table created successfully.");

} else {

System.out.println("' ProjectDetails ' table already exists.");

}

conn.close();

} catch (Exception e) {

e.printStackTrace();

}

}

public static boolean authenticateproject(String teamid) {

try {

Connection conn = DBconnect();

System.out.println("Connected Successfully");

String query = "select teamID from ProjectDetails where teamID = ?";

PreparedStatement pstmt = conn.prepareStatement(query);

pstmt.setString(1,teamid);

ResultSet rs = pstmt.executeQuery();

String searchedteamid = null;

while (rs.next()) {

searchedteamid = rs.getString(1);

}

conn.close();

return searchedteamid != null && searchedteamid.equals(teamid);

} catch (Exception e) {

e.printStackTrace();

}

return false;

}

static Connection conn = null;

public static Connection DBconnect(){

try{

String url = "jdbc:mysql://localhost:3306/";

String db = " finalyearprojectDB "; // database name

String username = "root";

String password = "Tanuvinnu@1234";

conn = DriverManager.getConnection(url+db, username, password);

}catch(Exception e){

e.printStackTrace();

}

return conn;

}

public static void searchbyprojectname(String projectName) {

try {

Connection conn = DBconnect();

System.out.println("Connected Successfully");

String query = "SELECT \* FROM projectDetails WHERE projectname = ?";

PreparedStatement pstmt = conn.prepareStatement(query);

pstmt.setString(1, projectName);

ResultSet rs = pstmt.executeQuery();

if (!rs.isBeforeFirst()) {

System.out.println("No project Details found for project: " + projectName);

} else {

System.out.printf("%-10s | %-50s | %-20s | %-20s |%-20s |%-20s |%-20s | %-20s%n",

"teamId","projectname","leadername", "teammember1"," teammember2", "teammember3","year","researchpaperstatus");

System.out.println("-----------------------------------------------------------------------------");

while (rs.next()) {

String teamId = rs.getString("teamId");

String leadername = rs.getString("leadername");

String projectname = rs.getString("projectname");

String teammember1= rs.getString("teammember1");

String teammember2 = rs.getString("teammember2");

String teammember3 = rs.getString("teammember3");

String year = rs.getString("year");

String researchpaperstatus = rs.getString("researchpaperstatus");

System.out.printf("%-10s | %-50s | %-20s | %-20s |%-20s |%-20s | %-20s | %-20s%n",

teamId,projectname,leadername, teammember1, teammember2, teammember3 ,year,researchpaperstatus);

}

}

conn.close();

} catch (Exception e) {

e.printStackTrace();

}

}

public static void searchbystudentname(String studentname) {

try {

Connection conn = DBconnect();

System.out.println("Connected Successfully");

String query = "SELECT \* FROM projectDetails WHERE leadername = ? OR teammember1 = ? OR teammember2 = ? OR teammember3 = ?";

PreparedStatement pstmt = conn.prepareStatement(query);

pstmt.setString(1, studentname);

pstmt.setString(2, studentname);

pstmt.setString(3, studentname);

pstmt.setString(4, studentname);

ResultSet rs = pstmt.executeQuery();

if (!rs.isBeforeFirst()) {

System.out.println("No project Details found in which either of the team member is with name: " +studentname);

} else {

System.out.printf("%-10s | %-50s | %-20s | %-20s |%-20s |%-20s |%-20s | %-20s%n",

"teamId","projectname","leadername", "teammember1"," teammember2", "teammember3","year","researchpaperstatus");

System.out.println("-----------------------------------------------------------------------------");

while (rs.next()) {

String teamId = rs.getString("teamId");

String leadername = rs.getString("leadername");

String projectname = rs.getString("projectname");

String teammember1= rs.getString("teammember1");

String teammember2 = rs.getString("teammember2");

String teammember3 = rs.getString("teammember3");

String year = rs.getString("year");

String researchpaperstatus = rs.getString("researchpaperstatus");

System.out.printf("%-10s | %-50s | %-20s | %-20s |%-20s |%-20s | %-20s | %-20s%n",teamId,projectname,leadername, teammember1, teammember2, teammember3 ,year,researchpaperstatus);

}

}

conn.close();

} catch (Exception e) {

e.printStackTrace();

}

}

public static void viewallprojects () {

try {

Connection conn = DBconnect();

System.out.println("Connected Successfully");

String query = "SELECT \* FROM projectDetails ";

PreparedStatement pstmt = conn.prepareStatement(query);

ResultSet rs = pstmt.executeQuery();

if (!rs.isBeforeFirst()) {

System.out.println("No project Details found");

} else {

System.out.printf("%-10s | %-50s | %-20s | %-20s |%-20s |%-20s |%-20s | %-20s%n",

"teamId","projectname","leadername", "teammember1"," teammember2", "teammember3","year","researchpaperstatus");

System.out.println("-----------------------------------------------------------------------------");

while (rs.next()) {

String teamId = rs.getString("teamId");

String leadername = rs.getString("leadername");

String projectname = rs.getString("projectname");

String teammember1= rs.getString("teammember1");

String teammember2 = rs.getString("teammember2");

String teammember3 = rs.getString("teammember3");

String year = rs.getString("year");

String researchpaperstatus = rs.getString("researchpaperstatus");

System.out.printf("%-10s | %-50s | %-20s | %-20s |%-20s |%-20s | %-20s | %-20s%n",teamId,projectname,leadername, teammember1, teammember2, teammember3 ,year,researchpaperstatus);

}

}

conn.close();

} catch (Exception e) {

e.printStackTrace();

}

}

}

**Faculty.java**

package Projpack;

import java.sql.Connection;

import java.sql.PreparedStatement;

import java.util.HashSet;

import java.util.Random;

import java.util.Scanner;

import java.util.Set;

public class Faculty{

public static void addDetails(String teamID,String projectname,String leadername,String teammember1,String teammember2,String teammember3,String year,String rpstatus)

{

try {

Connection conn = FYproject.DBconnect();

System.out.println("Connected Successfully");

// Check if the project already exists

if(FYproject.authenticateproject(teamID)) {

System.out.println("Project with this ID already exists. Cannot add again.");

conn.close();

return;

}

// Add the project

String addProjectQuery = "INSERT INTO ProjectDetails(teamID,projectname,leadername,teammember1,teammember2,teammember3,year,researchpaperstatus) VALUES (?,?,?,?,?,?,?,?)";

PreparedStatement addProjectStmt = conn.prepareStatement(addProjectQuery);

addProjectStmt.setString(1,teamID);

addProjectStmt.setString(2,projectname);

addProjectStmt.setString(3,leadername);

addProjectStmt.setString(4,teammember1);

addProjectStmt.setString(5,teammember2);

addProjectStmt.setString(6,teammember3);

addProjectStmt.setString(7,year);

addProjectStmt.setString(8,rpstatus);

addProjectStmt.executeUpdate();

System.out.println("Project added successfully");

conn.close();

}catch(Exception e) {

e.printStackTrace();

}

}

public static void deleteProject(String teamID) {

try {

Connection conn = FYproject.DBconnect();

System.out.println("Connected Successfully");

// Check if the project already exists

if(FYproject.authenticateproject(teamID)) {

System.out.println("Project with this ID exists. Can be deleted.");

// Delete a project

String deleteProjectQuery = "DELETE FROM projectDetails WHERE teamID = ?";

PreparedStatement deleteProjectStmt = conn.prepareStatement(deleteProjectQuery);

deleteProjectStmt.setString(1,teamID);

deleteProjectStmt.executeUpdate();

System.out.println("Project deleted Successfully.");

conn.close();

}

}catch(Exception e) {

e.printStackTrace();

}

}

public static void update(String researchpaperstatus,String teamID) {

try {

Connection conn = FYproject.DBconnect();

System.out.println("Connected Successfully");

// Check if the project already exists

if(FYproject.authenticateproject(teamID)) {

System.out.println("project with this ID exists. Can be updated.");

// Update the research paper status of the project

String updateProjectQuery = "UPDATE projectDetails SET researchpaperstatus = ? WHERE teamID = ?";

PreparedStatement updateProjectStmt = conn.prepareStatement(updateProjectQuery);

updateProjectStmt.setString(1,researchpaperstatus);

updateProjectStmt.setString(2,teamID);

updateProjectStmt.executeUpdate();

System.out.println("research paper status of the Project updated successfully");

conn.close();

}

}catch(Exception e) {

e.printStackTrace();

}

}

}